Memory and Unicode: Takeaways

Syntax

- Converting a number to binary:

```
int("100", 2)
```

- Encoding a string into bytes:

```
batman.encode("utf-8")
```

- Converting a string of length one to a Unicode code point:

```
ord("a")
```

- Converting integer to binary string:

```
bin(100)
```

- Decoding a bytes object into a string:

```
morgan_freeman.decode()
```

- Counting how many times a string occurs in a list:

```
from collections import Counter

fruits = ["apple", "apple", "banana", "orange"]

fruit_count = Counter(fruits)
```

Concepts

- Hard drives are referred to as magnetic storage because they store data on magnetic strips.
- Magnetic strips can only contain a series of two values — ups and downs.
- Using binary, we can store strings in magnetic ups and downs. In binary, the only valid numbers are `0` and `1` so it's easy to store binary values on a hard drive.
- Binary is referred to as base two because there are only two possible digits — 0 and 1. We refer to digits as `0-9` as base 10 because there are 10 possible digits.

- We can add numbers in binary like we can in base 10.
- Computers store strings in binary. Strings are split into single characters and then converted to integers. Those integers are then converted to binary and stored.
- ASCII characters are the simple characters — upper and lowercase English letters, digits, and punctuation symbols. ASCII only supports 255 characters.
- As a result of ASCII's limitation, the tech community adopted a new standard called Unicode. Unicode assigns "code points" to characters.
- An encoding system converts code points to binary integers. The most common encoding system for Unicode is UTF-8.
- UTF-8 supports all Unicode characters and all ASCII characters.
- Bytes are similar to a string except that it contains encoded byte values.
- Hexadecimal is base 16. The valid digits in hexadecimal are `0-9` and `A-F`.
    - A: 10
    - B: 11
    - C: 12
    - D: 13
    - E: 14
    - F: 15
- Hexadecimal is used because it represents a byte efficiently. For example, the integer 255 in base 10 is represented by 11111111 in binary while it can be represented in hexadecimal using the digits FF.

Resources

- Binary numbers
- Unicode